View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Elsevier - Publisher Connector

Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Biomedical Informatics 37 (2004) 305–318
Journal of Biomedical Informatics
www.elsevier.com/locate/yjbin

Design and implementation of the GLIF3 guideline execution engine

Dongwen Wang,[a,*] Mor Peleg,[b,d,1] Samson W. Tu,[b] Aziz A. Boxwala,[c,e,1] Omolola Ogunyemi,[c] Qing Zeng,[c] Robert A. Greenes,[c] Vimla L. Patel,[a] and Edward H. Shortliffe[a]

[a] *Department of Biomedical Informatics, Columbia University, New York, NY 10032, USA*
[b] *Stanford Medical Informatics, Stanford University, Stanford, CA 94305, USA*
[c] *Decision Systems Group, Brigham and Women's Hospital, Harvard Medical School, Boston, MA 02115, USA*
[d] *Department of Management Information Systems, University of Haifa, Haifa 31905, Israel*
[e] *Eclipsys Corporation, Boston, MA 02135, USA*

Received 20 June 2003
Available online 20 July 2004

Abstract

We have developed the GLIF3 Guideline Execution Engine (GLEE) as a tool for executing guidelines encoded in the GLIF3 format. In addition to serving as an interface to the GLIF3 guideline representation model to support the specified functions, GLEE provides defined interfaces to electronic medical records (EMRs) and other clinical applications to facilitate its integration with the clinical information system at a local institution. The execution model of GLEE takes the "system suggests, user controls" approach. A tracing system is used to record an individual patient's state when a guideline is applied to that patient. GLEE can also support an event-driven execution model once it is linked to the clinical event monitor in a local environment. Evaluation has shown that GLEE can be used effectively for proper execution of guidelines encoded in the GLIF3 format. When using it to execute each guideline in the evaluation, GLEE's performance duplicated that of the reference systems implementing the same guideline but taking different approaches. The execution flexibility and generality provided by GLEE, and its integration with a local environment, need to be further evaluated in clinical settings. Integration of GLEE with a specific event-monitoring and order-entry environment is the next step of our work to demonstrate its use for clinical decision support. Potential uses of GLEE also include quality assurance, guideline development, and medical education.
© 2004 Elsevier Inc. All rights reserved.

Keywords: Clinical practice guidelines; Knowledge representation; Theoretical models; Clinical decision support systems; Computer-assisted decision-making; Health care quality assurance; Guideline execution engine

1. Introduction

In recent years, computer-based guideline models have been developed to address the need for guideline representation and execution [1,2]. These models are used as generic templates to facilitate the translation of guidelines from their published formats into computer-interpretable algorithms [3], to share the clinical knowledge embedded within guidelines [4], and to assist the integration of guidelines with the clinical information system at a local institution to provide patient-specific clinical decision support [5].

Different approaches have been used in the previously developed guideline execution engines, the primary function of which is to interpret and to execute the guidelines encoded in specific representation formats [6–10]. These approaches usually provided a standard interpretation of the encoded guidelines, with the guideline execution engines as the interpreters of specific guideline encoding languages. Two important issues that have not been addressed appropriately by these previous

* Corresponding author. Fax: 1-212-305-3302.
E-mail address: dongwen.wang@dbmi.columbia.edu (D. Wang).
[1] Mor Peleg was at Stanford University when working on this project and is now at University of Haifa. Aziz Boxwala was at Brigham and Women's Hospital when working on this project and is now at Eclipsys Corporation.

1532-0464/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jbi.2004.06.002

approaches are: (1) the flexibility of guideline execution, such as the handling of the possible disagreement by a clinician on the guideline recommendations generated by a computer system [11], and (2) the support for the maintenance of the guideline execution engine when the guideline representation model evolves over time.

In 1998, our group, the InterMed Collaboratory (a research consortium created by informatics groups from Columbia University, Harvard University, and Stanford University), published the GuideLine Interchange Format (GLIF), aiming to use it as a standard representation model for the sharing of guidelines among different institutions [12]. Later, a prototype guideline execution engine was designed and implemented to integrate with the clinical information system at a local institution for the execution of guidelines encoded in an enhanced format of the second version of GLIF (GLIF2) [10]. In response to that experience, the limitations of GLIF2, such as the ad hoc approach to the definition of patient data and clinical actions, the lack of a specification for the logical expressions, and the limited set of decision models, have been addressed, and new requirements for guideline modeling, such as the representation of a patient's clinical state, have been included, resulting in the third version of GLIF (GLIF3) [13]. In this paper, we present our approach to the design and implementation of the GLIF3 Guideline Execution Engine (GLEE) that tries to balance the requirements of the shareability of guideline encoding, the flexibility of guideline execution, and the maintainability of guideline implementation tool. Although GLEE has not yet been integrated with a functioning clinical system, we believe that its design, implementation, and testing as a functioning modular tool provide useful intermediate lessons for the biomedical informatics community.

2. Overview of the GLIF3 guideline representation model

To help in understanding the GLEE functions that we describe in the subsequent sections, we provide here a brief overview of the GLIF3 guideline representation model. Detailed specification of the GLIF3 guideline representation model can be found elsewhere [14].

In the GLIF3 model, guidelines are represented as specific *guideline* instances. The process of clinical care is encoded as the *algorithm* of a guideline. Within an algorithm, instances of five types of tasks, which are called *guideline steps*, can be encoded and linked together in a flowchart to specify their scheduling and coordination during guideline application. Specifically, *action steps* are used to record clinical or computational actions; *decision steps* are used to encode decision points; *patient state steps* are used to specify a patient's pathophysiological or management states in the specific contexts of a guideline's application; and *branch steps* and *synchronization steps* are used to schedule and to coordinate concurrent tasks or tasks with arbitrary execution order. The clinical care process represented in the GLIF3 model can be nested using *subguidelines*, and thus multiple views to the care process with different granularities can be defined. Clinical data in GLIF3 are encoded as *data items*. These data items are then referenced by *expressions*, which are used to encode *decision criteria* and *patient state*. Clinical events in GLIF3 are encoded as *triggering events*, which are used to activate specific clinical tasks.

GLIF is a guideline representation model that continues to evolve. The GLIF3 guideline representation model used for this research, represented in the Resource Description Framework (RDF) format [15], can be found at: http://guidelines.dbmi.columbia.edu/GLEE/GLIF.rdfs.

3. Philosophy of design

GLEE is an important component in InterMed's framework of guideline sharing [2]. It is built as middleware that is intended to be integrated with the clinical information system at a local institution through defined interfaces to its electronic medical records (EMRs) and clinical applications. In addition to clinical decision support, GLEE aims to be used for quality assurance, guideline development, and medical education [16]. GLEE is currently implemented using the JAVA programming language.

3.1. GLEE and guideline lifecycle

It is InterMed's vision that the lifecycle of a computer-interpretable guideline consists of a series of stages: (1) conceptual modeling, (2) encoding, (3) validation, (4) dissemination, (5) local adaptation, (6) integration with an implementation system, and (7) application and revision [2]. In this lifecycle, GLEE is to be used primarily during the guideline implementation stages (6) and (7). Specifically, GLEE provides defined interfaces to the clinical information system at a local institution with a goal that guideline implementation integrates seamlessly with the local environment. Since GLEE can be linked to EMRs, it is intended to be used to assist in the application of guidelines to specific patients. In addition, as GLEE can be used as a tool to assist in guideline development through iterative refinement, it is related to the conceptual modeling, encoding, and validation of a guideline. The stages of a guideline's lifecycle and GLEE's role in these stages are shown in Fig. 1.

3.2. GLEE and the GLIF3 guideline representation model

To share the medical knowledge encoded in a specific guideline, the guideline represented in the GLIF3 format

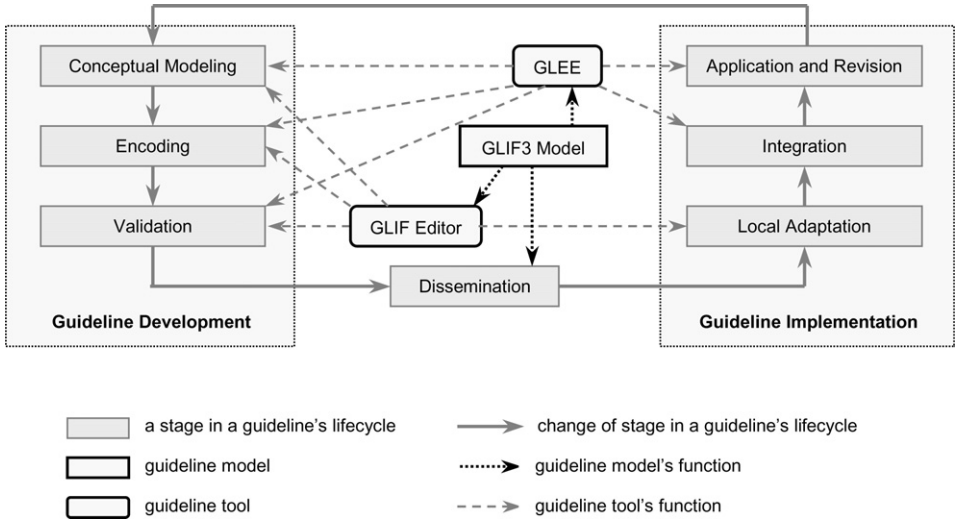

Fig. 1. InterMed's vision on a guideline's lifecycle and its relationship to guideline model and tools. GLIF editor is used primarily during the guideline development stages. It can also be used for local adaptation of guidelines. GLEE is used primarily during the guideline implementation stages. It can also be used as an assisting tool during the guideline development stages. The whole process is based on the GLIF3 guideline representation model.

needs to be interpreted correctly. This can be realized at a local institution through an ad hoc implementation, where an institution-specific computer program is written to interpret and to integrate that particular guideline. Obviously, a local institution that takes this approach to guideline implementation needs to invest extensive resources. Alternatively, implementation of a guideline that is encoded in the GLIF3 format can be based on a consistent approach to the interpretation, integration, and application of the GLIF3 model, which acts as a template of any GLIF3 guidelines during this process. This standard approach to the execution of a guideline is an important requirement for guideline sharing [17], and GLEE was developed precisely for this purpose. In this paper, we accordingly describe GLEE as a modular tool that properly applies the GLIF3 model during guideline execution. As such, it serves as a reference for implementation of a GLIF3 execution engine that others can use. GLEE's integration with specific clinical systems is the subject of further work.

4. System architecture

Since our goal was to develop GLEE as a tool that would be taken by a local institution to integrate with its clinical information system, the system architecture of GLEE was designed to provide flexibility for such integration.

4.1. GLEE and host clinical information systems

GLEE provides interfaces intended to support integration with the host clinical information system at a local institution. These interfaces are used to link GLEE to a local EMR at the back-end and associated clinical applications (e.g., a physician order-entry system) at the front-end. The communication between GLEE and the EMR at the back-end will enable GLEE's access to various resources in the local environment, such as retrieval of patient data and monitoring of clinical events in case the local institution needs to trigger a guideline through specific clinical events. The communication between GLEE and associated clinical applications at the front-end is intended to enable smooth integration of the decision support services provided by GLEE, such as alerts and reminders, within a clinician's workflow [18]. In other words, GLEE defines the business logic of a guideline application, the local EMR will provide data, and the associated clinical applications will support the interactions between users and a guideline implementation system. The overall system architecture is shown in Fig. 2. As the execution engine for GLIF3 guidelines, GLEE supports the functions defined in the GLIF3 model. However, it is important to note that not all the components in Fig. 2 would be required for GLEE to work properly. For example, if a local institution does not have a clinical event monitor and the guidelines implemented at that institution are not triggered by clinical events, GLEE can function properly without the integration with a clinical event monitor.

4.2. Internal structure of GLEE

Internally, GLEE takes a layered approach to partitioning the functions provided by its components. GLEE is built as a client-server system to obtain maxi-

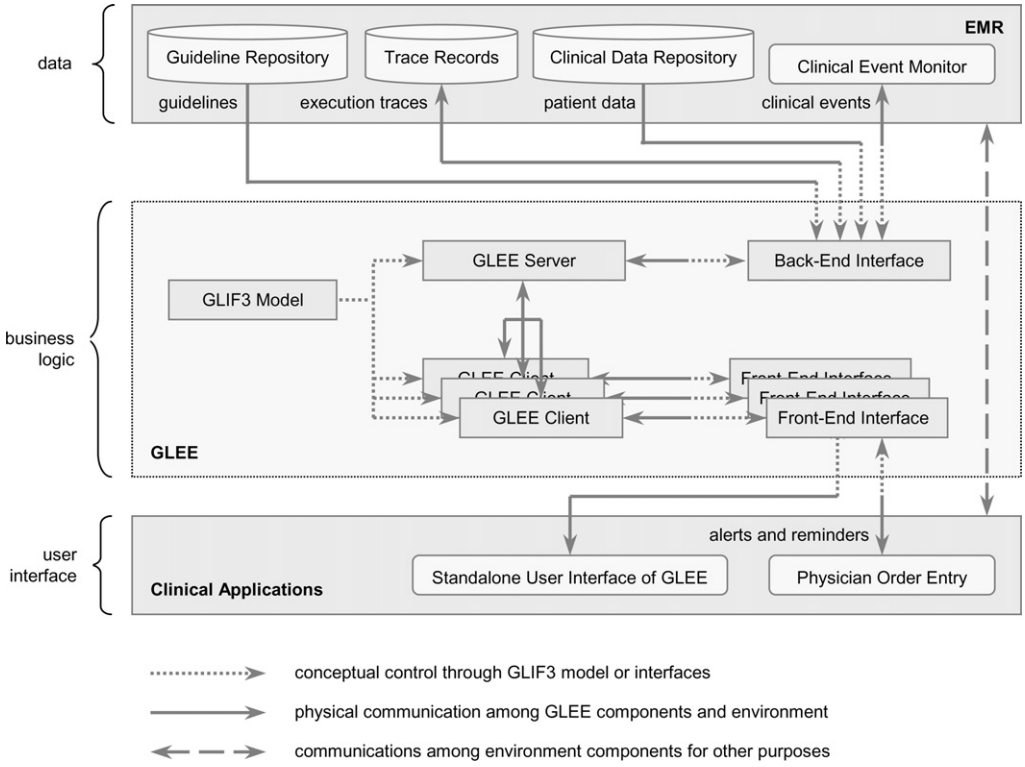

Fig. 2. The internal structure of GLEE and its interactions with a local environment. GLEE's components can be classified into three conceptual layers, the GLIF3 model, the core components, and the interfaces to the host environment. The communication between GLEE and the EMR at the back-end enables GLEE's access to various resources in the local environment, such as retrieval of patient data, and monitoring of clinical events. The communication between GLEE and associated clinical applications at the front-end enables smooth integration of the decision support services provided by GLEE, such as alerts and reminders, within a clinician's workflow. The EMR, GLEE, and associated clinical applications define the data, the business logic, and the user interface for a guideline's application. Multiple GLEE clients can be instantiated simultaneously. The standalone user interface is used only for development and demonstration purposes.

mum flexibility in its integration with host systems. In addition to the interfaces to a local clinical information system, GLEE also provides a standalone user interface to facilitate development, testing, and demonstration.

4.2.1. Three-layer conceptual structure

GLEE's components can be classified into three conceptual layers: (1) the GLIF3 guideline representation model, (2) the core components of GLEE, and (3) the interfaces to a host clinical information system. The GLIF3 guideline representation model specifies a set of generic functions, such as recommendations for specific clinical actions and assistance in medical decision-making, which should be supported by any tool executing guidelines encoded in the GLIF3 format. The core components of GLEE, as an execution environment for GLIF3, define an execution model to realize the generic functions that are required by the GLIF3 representation model. The interfaces to a host clinical information system reflect GLEE's assumptions on the interactions between GLEE and its host environment during guideline execution. The internal structure of GLEE and the relationships among its three conceptual layers are shown in Fig. 2.

As the interfaces between the GLIF3 guideline representation model and the core components of GLEE are clearly defined, maintenance of GLEE is facilitated by this component-based approach to its development. Since GLIF is an evolving guideline representation model, this approach can facilitate future enhancement of GLEE when new versions of the GLIF model are developed and thus additional features of the model need to be supported by GLEE.

4.2.2. Client-server system

Flexibility in the integration of GLEE with a local clinical information system and its associated clinical applications is a primary concern in GLEE's design. Since a guideline can be applied to multiple patients at the same time and a patient can be simultaneously eligible for multiple compatible guidelines at a specific moment, efficient processing of these interwoven one-to-many and many-to-one relationships is important during guideline execution. We accordingly chose to build GLEE as a client-server system, with each GLEE client corresponding to the application of a guideline to a particular patient. Specifically, a GLEE server is developed (1) to handle the interface with the GLIF3 model

and to regenerate the internal structure of the guideline so as to obtain its correct interpretation, (2) to process the communication with a local environment at the back-end (including retrieval of guidelines from a guideline repository, reading and writing of execution traces when a guideline is applied to a specific patient, accessing of patient data from a clinical data repository, and monitoring of possible clinical events that are used to trigger the execution of a guideline), (3) to manage the clients (including bookkeeping of the guideline and patient in a specific GLEE client, and recording the information for client-server communication), and (4) to provide computational support for task scheduling, task execution, and state transition for a specific client. Meanwhile, a GLEE client is developed (1) to support the interactions between a clinical application and a user when that clinical application uses the services provided by GLEE (such as recommendations of clinical actions and assistance with medical decision-making as defined in GLIF3 guidelines), and (2) to record the execution state in a specific round of a guideline's application to a patient. The relationship between a GLEE server and its clients as well as their functions in the overall system architecture of GLEE are shown in Fig. 2.

4.2.3. Standalone user interface

Although GLEE's execution will ultimately be invoked by running specific clinical applications, we have built a standalone user interface at the client side. As shown in Fig. 3, this standalone user interface is used to present to developers and implementers the process structure of a GLIF3 guideline as well as the active steps at a specific moment when that guideline is being applied to a testing patient. It can also be used to check the detailed information regarding a guideline step. Users can interact with GLEE using this client-side standalone user interface to decide whether to start, continue, or stop the execution of a specific guideline step. They can also find the documentation about the guideline, such as the references to the original published guideline and the maintenance information about the encoded guideline. It is important to note that this standalone user interface is developed for system development, debugging, and demonstration purposes rather than to be used directly by clinicians in practice. In this paper, we use the standalone user interface to illustrate the function of GLEE. We expect that this standalone user interface, with appropriate enhancements of its function, can be adapted in the future for medical education to

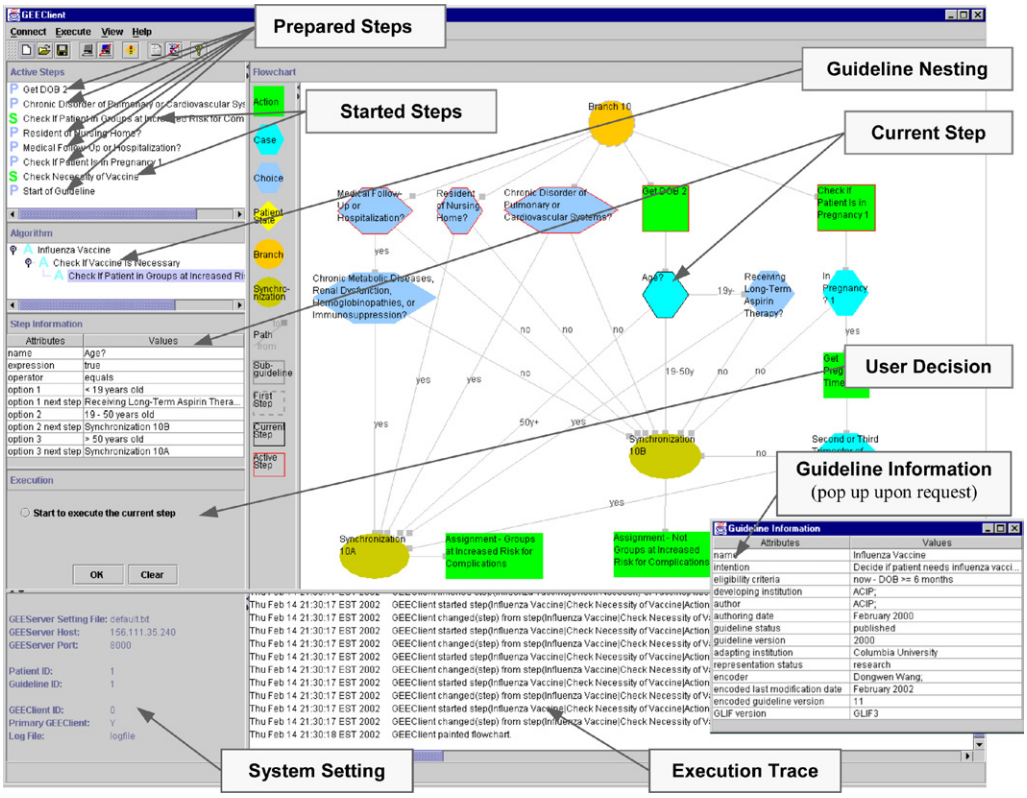

Fig. 3. A screenshot of the standalone user interface at the client side of GLEE during development and testing. The algorithm of a GLIF3 guideline is shown as a flowchart at the upper-right portion of the screen. A list of active steps, the hierarchy of algorithms, and detailed information on the currently highlighted step are shown in the upper-left portion of the screen. The setting of the current client and the execution trace are shown in the lower portion of the screen. Maintenance information of the guideline is shown in the pop-up window. Note that this user interface is intended to be used by developers for testing and demonstration rather than by clinicians for patient care.

disseminate the clinical knowledge encoded in a guideline and as a testing tool augmenting the function of a guideline editor to validate the encoded guidelines. A guideline's presentation to a clinician providing patient care will be very different, however, and will generally require tight integration of GLEE with other clinical systems such as order-entry or result-reporting.

4.3. System interfaces

In the system architecture of GLEE, guidelines encoded in the GLIF3 format are stored in a guideline repository, from which they can be retrieved by the GLEE server. The system interface between GLEE and the guideline repository is defined for this purpose. As retrieval of a guideline occurs through this defined interface, GLEE does not make any assumption on how the guidelines are stored in the guideline repository. However, as GLEE needs to parse a guideline to create the internal representation of that guideline when it is retrieved from the guideline repository for the first time, the syntax of the retrieved guideline should be predefined. Currently, the GLIF3 guidelines we have used were developed using the Protégé-2000 knowledge acquisition tool [19] and exported as RDF files [15].

The system interface between the GLEE server and a host clinical information system at the back-end is defined for several purposes. First, the GLEE server's retrieval of patient data from the local clinical data repository is through this system interface. Second, the GLEE server's registration of clinical events with the local clinical event monitor, and the local clinical event monitor's notification to the GLEE server on triggering of specific clinical events, are through this system interface. In addition, this system interface is used for the retrieval of execution traces, which record the history of the application of a specific guideline to a particular patient, and for sending back to the host system the trace records that document the recently performed executions. Finally, a messaging function, which is provided to support the GLEE server's sending of a generic message to the host system for specific types of communication (the semantics can be defined at a local institution), is also implemented through this system interface between the GLEE server and the host clinical information system.

The system interface between the GLEE client and the clinical applications at the front-end is defined to facilitate the interactions between a user and GLEE. It is used (1) to select a particular guideline and a specific patient for execution, (2) to provide recommendations for clinical actions, (3) to assist medical decision-making, (4) to verify a patient's clinical or management state, (5) to execute a guideline at different granularity levels of its subguidelines, and (6) to support a user's subjective decision on guideline execution (i.e., their decision on whether to follow the guideline's advice or to pursue a different course). This system interface is also used by GLEE's client-side standalone user interface to support the interactions between a developer user and GLEE. For example, at a specific decision point of a guideline,[2] a GLEE client first sends the set of possible options to the user interface so that they can be presented. Once the decision has been made, the result of the decision is sent back to the GLEE client. In a clinical application, such as a physician order-entry system, if the decision is to select an appropriate medication for a specific patient, a GLEE client needs to send to the order-entry system the set of medications that can be selected in that context. The order-entry system then sends back to the GLEE client the specific medication that has been ordered.

5. Task scheduling and tracing system

As described previously, a discrepancy may arise between a patient's expected state as encoded in a specific context of a guideline and his or her actual state as judged by a clinician during guideline application. GLEE thus allows a user to override a system's recommendation during guideline execution. To provide this flexibility in guideline execution, we developed an execution model for GLEE to support user-controlled task scheduling. In addition, we developed a tracing system for GLEE to record the guideline execution process. This tracing system, integrated with GLEE's guideline execution model, can be used to recover the execution history of a guideline's application to a specific patient.

5.1. User-controlled task scheduling

The traditional approach to task scheduling during guideline execution is to determine mechanically the executable tasks defined by an encoded guideline in a specific context when the guideline is applied to a patient. In this approach, the whole process of task scheduling is completely controlled by the execution engine. A major drawback of this approach is that an encoded guideline generally cannot address every possible clinical scenario, and thus may lead to discrepancies between what the guideline system suggests and what the clinicians may correctly determine should be done for a patient. Several guideline representation models, including GLIF3, have addressed this issue by providing representation primitives such as patient scenarios or patient states that can be used to record a

[2] In the standalone user interface, this is reflected as the start of a decision step. We provide detailed description on a guideline step's execution state in Section 5.

patient's clinical status in a specific context of a guideline [14,20,21]. These patient scenarios or patient states can then be used as entry/exit points to a guideline when applied to a patient. Although this solution can provide some level of flexibility in execution, it still depends to a large extent on the guideline encoders' enumeration of all possible entry/exit points for a guideline. This explains the need to allow a user to override the system's recommendation, which is important for the successful application (and acceptance) of a guideline in a clinical environment. We therefore decided that GLEE should provide an extra level of flexibility in guideline execution, with the user of GLEE as the final decision maker in task scheduling. In other words, at any time during the execution of a guideline, users can follow the task schedule suggested by the system, or they can start or stop the execution of any step based on their own judgment.

5.2. Execution states and transitions

To distinguish the steps scheduled by GLEE according to the encoded guideline from those actually executed according to a user's decision, we use four *execution states* to represent the status of a guideline step during execution. These four execution states of a guideline step include (1) the *prepared state*, which means a step is suggested as executable by the execution engine according to the encoded guideline, (2) the *started state*, which means a step has actually been started by a user, (3) the *stopped state*, which means a step has been intentionally stopped by a user before it starts or completes its execution, and (4) the *finished state*, which means a step has normally completed its execution based on GLEE's scheduling. The prepared state and the started state are jointly called the *active state*. The finished state and the stopped state are jointly called the *inactive state*. At any time during a guideline's execution, the execution states of its guideline steps are kept in an internal record. These execution states are updated appropriately during the guideline execution process.

Typically, the GLEE task scheduler suggests executable steps based on the scheduling information encoded in specific guideline steps, or, in the case of a new encounter when applying a specific guideline to a particular patient, based on the trace record of previous encounters with the guideline by the current patient. These executable steps are then put into the prepared states. Users can either confirm GLEE's suggestion on the execution schedule, or they can decide to override it by stopping a prepared step and starting another step they think to be appropriate. Users can also stop a started step to avoid unnecessary waiting for completion of an execution that is no longer relevant. If there is no manual interference from users, GLEE will decide when a started step should finish its execution. Usually, this will trigger the execution of other steps that will be put into the prepared states by the GLEE task scheduler.

We use an example to illustrate the change of a guideline step's state during the guideline execution process. Suppose GLEE has been used in the implementation of an immunization guideline to provide reminders of vaccines due for specific patients. Patient A has had two vaccine doses previously, but only the first dose has been recorded in the clinical data repository. During a visit of patient A, when the immunization guideline is invoked, GLEE puts the *1 Previous Dose* patient state step, which means the patient had one vaccine dose previously, into the prepared state according to the immunization history data stored in the clinical data repository. If this recommendation were confirmed by a clinician, the *1 Previous Dose* patient state step would be put into the started state such that the system would be able to check the patient's eligibility for the 2nd dose of the vaccine. However, if patient A's physician finds from a paper record that patient A has already had two previous vaccine doses, she will refuse the recommendation generated by the system. Accordingly, GLEE will put the *1 Previous Dose* patient state step into the stopped state and put the *2 Previous Doses* patient state step, which means the patient has two vaccine doses previously, into the started state (in a functioning clinical system this can be implemented through the documentation or re-entry of the missing dose) such that the system can check the patient's eligibility for the 3rd dose of the vaccine. In this way, a clinician user will be able to correct the inappropriate recommendations that may be generated by the GLEE system. The execution states of a guideline step and the transitions between these states are shown in Fig. 4.

Providing execution flexibility by introducing users' decisions regarding task scheduling does not mean that users have to make this decision in each and every step of guideline execution. By appropriate configuration, the system's automatic execution (for those tasks with clearly defined and verified logic) and users' subjective decisions (for those tasks that need clinicians' judgments) can be combined to serve special needs when implementing a particular guideline. For example, we currently provide a *batch execution mode* to support the application of a specific guideline to multiple patients. When running in this mode, GLEE automatically accepts all executable steps recommended by its task-scheduler, and selects a prepared step to execute each time a user selection is needed. This batch execution mode was used in the evaluation of GLEE (described in Section 10).

5.3. Tracing system

Keeping the trace of execution for a specific guideline when it is applied to a particular patient is an important

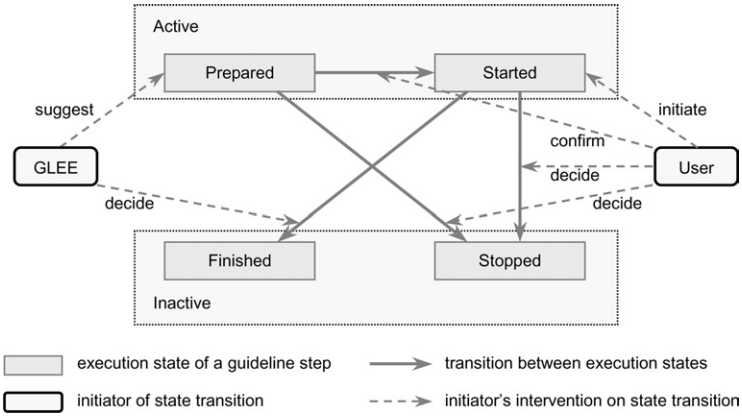

Fig. 4. The execution state of a guideline step and the transitions between these execution states. GLEE suggests an executable step and puts it into the prepared state. A user of GLEE decides whether to follow GLEE's suggestion or to deny it by stopping the suggested step and initiating the start of another step. A user can also stop a started step that is no longer relevant. If there is no manual interference from users, GLEE decides when a started step should finish its execution.

feature of GLEE. Specifically, GLEE creates the guideline execution trace of a patient to record (1) the activation of a step, (2) the start of a step, (3) the finish of a step, (4) the stop of a step, and (5) the chaining of steps. When implemented as an integrated component of an EMR environment, these patient-specific traces would be stored internal to the GLEE tool as described below.

The execution trace for a specific patient can be used as a hint for task scheduling in future encounters with the patient when the same guideline is re-applied. Specifically, if the trace has shown that the execution of specific guideline steps had not been finished in previous applications of the guideline to the current patient, those unfinished guideline steps will be recommended by GLEE as the potential starting points when the same guideline is re-applied to that patient. This feature of GLEE is provided for implementation of guidelines on chronic disease management, where the application of a guideline to a particular patient usually involves multiple encounters. With GLEE's capability to parse the execution trace that records previous encounters, application of a guideline to a specific patient can start from a particular point in the guideline that corresponds to the patient's current management state instead of from the beginning of the guideline in each encounter.

The execution trace can provide a complete record of a guideline's execution for quality assurance purposes to determine whether the care provided to a specific patient is in compliance with guideline recommendations. Ideally, the trace record used for such purposes would simply be extracted from a patient's medical record. In fact, some of the execution history that is recorded in the trace (e.g., the start of an action step that corresponds to a specific type of clinical intervention, such as the prescription of a medicine) can be found in a typical medical record. However, many other elements in GLEE's trace records may not have been documented there but are required for the proper execution of a guideline. For this reason, we chose to keep an independent trace record system that could be used by the GLEE server, as shown in Fig. 2. Currently, these trace records are implemented as XML [22] files that are stored at the server side of GLEE. The document type definition (DTD) of the execution trace XML file can be found at: http://guidelines.dbmi.columbia.edu/GLEE/GLEETraceDTD.html.

6. Event model

An event-driven execution model is an intrinsic part of many clinical decision support systems [23]. GLIF3 supports such an execution model by defining triggering events for a specific guideline step.

The support for the event-driven guideline execution mode depends on the availability of the clinical event monitor at a local institution. If a local institution would like to support the guideline execution mode that is driven by clinical events, such as the newly arrived lab results or the entry of physician orders, it needs to provide a clinical event monitor with which the GLEE server connects. During guideline execution, when a step with a triggering event is started, GLEE first registers that event with the clinical event monitor at the local site. It then waits for the occurrence of the registered event to trigger its execution. Once a clinical event occurs, the clinical event monitor at the local institution sends a message to the GLEE server to notify the triggering of the event. The guideline step that is waiting for the event is then triggered to start its execution.

7. Task execution

Execution of the specific types of tasks defined in a guideline representation model is one of the most important functions of a guideline execution engine. For

GLEE, this means the handling of the representation elements in GLIF3 that are used to encode the specific types of clinical tasks as well as those that are used to support the scheduling of the clinical tasks.

7.1. Execution of clinical tasks

In the GLIF3 guideline representation model, the elements that are used to represent the different types of clinical tasks include *action step*, *decision step*, and *patient state step*.

Handling of an action step depends on the type of task defined in that step. Specifically, GLEE sends a message to notify the local clinical information system if the task defined in the action step is a *medically oriented action*; GLEE updates its *internal data assignment record* if the task is an *assignment action*; GLEE communicates with the clinical data repository at the local institution to retrieve specific patient data and updates the internal data assignment record if the task is a *get data action*; and finally, GLEE starts the execution of a subguideline if the task is a *subguideline action*. Once it has finished the processing of the tasks, GLEE obtains the subsequent step as defined in the current action step. This subsequent step is then scheduled to be executed.

The decision step in the GLIF3 model can be classified into *case step*, which represents a decision-making process that can be implemented by GLEE automatically, and *choice step*, which represents a decision-making process that needs inputs from a user. GLEE handles the case step and choice step differently. For a case step, GLEE scans its *options* and evaluates the *criterion* of the *case condition* in each of the options until the criterion of an option can be satisfied, leading to the selection of that option as the decision result. For a choice step, GLEE presents the *options* of the step to a user at the client side and waits for the user's decision on the selection of the options. After an option is selected in the decision-making process, the subsequent step corresponding to that option is obtained. GLEE then schedules this subsequent step to be executable and updates the relevant trace record.

The patient state step in the GLIF3 model is used as a label to specify a patient's clinical or management state in a particular context of a guideline's application. The *patient state description* of a patient state step is a criterion to define a patient's status as represented by the step (i.e., the eligibility/applicability criteria). Ideally, this information should be compared to a patient's actual state during guideline application to automatically select or narrow the search for the possible states of a patient. Currently, this information is simply presented to a user at the client side; the user will make the final decision on whether the observed patient data match with the criterion defined in the patient state step. If so, the patient state is validated and the subsequent step is scheduled.

7.2. Execution of scheduling tasks

In the GLIF3 guideline representation model, the elements that are used to represent the different types of scheduling tasks include *branch step*, *synchronization step*, and *subguideline*. These tasks constitute GLEE's computational support for workflow management so that coordination of specific tasks can be implemented during guideline application.

The branch step in the GLIF3 model is used to represent a diverging point in a guideline's algorithm so that concurrent tasks and tasks with arbitrary execution order can be represented. The branch step itself does not have any internal tasks that need to be performed. Its uniqueness is its chaining with the subsequent steps. The major difference between a branch step and other types of steps is that a branch step has multiple subsequent steps, while others have only one. Consequently, after the execution of a branch step, GLEE needs to schedule all the subsequent steps by putting them into the prepared state so that they will become executable.

The synchronization step in the GLIF3 model is used to represent a converging point in a guideline's algorithm so that multiple tasks can be coordinated during a guideline's application. When a synchronization step is executed, its *continuation* criterion is evaluated. During this evaluation, GLEE checks the execution history as recorded in the execution traces. If the *continuation* criterion is not satisfied, the synchronization step will continue to wait until the completion of other steps eventually leads to the fulfillment of the *continuation* criterion.

In the GLIF3 model, subguidelines are used to provide different views to the clinical care process. When a step with a subguideline defined is started, the user of GLEE can select to go to the lower level of the guideline hierarchy to execute the subguideline, or to skip the subguideline to keep the execution at the current level of the guideline hierarchy if the goal of the subguideline has already been achieved. Once a user decides to execute the subguideline, the first step of the subguideline is scheduled to be executable, leading to the initialization of that subguideline's execution. During the whole execution process of the subguideline, the step of the upper level guideline within which the subguideline is defined will remain in the started state. After the execution of an ending step, beyond which no subsequent step is defined in the subguideline, execution of the subguideline is finished, leading to the return of the control of the task scheduling back to the upper level (invoking) guideline.

8. Patient data retrieval, clinical event registration, and clinical action notification

Access to patient data is a critical task in guideline execution [24]. For guidelines to be shared across

different institutions, a standard data encoding system and a generic patient data model are two prerequisites. This standard data encoding system plus a generic patient data model will enable references to patient data in an encoded guideline such as in a specification of decision criteria without the need to know the implementation details. At a local institution, the standard definition of patient data are then mapped to the implementation-specific data schema and access methods of the local EMR. In recent years, several controlled medical terminologies, such as SNOMED-CT [25] and LOINC [26], have been developed as standards for patient data encoding. However, there is as yet little consensus on a common patient data model in the biomedical informatics research community. For this reason, we do not assume any specific standard on the use of controlled medical terminologies and patient data models. Instead, guideline encoders can select their own controlled medical terminology and patient data model when encoding patient data [27]. During guideline execution, patient data access is through a standard interface to the clinical data repository at a local institution, with the identification of the terminology, the concept in the terminology that is used to represent the data, the patient data model, and the specific data-model class as the parameters in the communication. Using these parameters and the mapping between this definition of patient data and the schema of the clinical data repository at a local institution, the patient data required by GLEE during guideline execution can then be retrieved from the local clinical data repository. This approach is compatible with the recent research on the development of a *virtual medical record* by building a standard patient data model on the basis of HL7's Reference Information Model (RIM), which is then mapped to the database schema of a local institution [28]. It is important to note that this way to use a controlled medical terminology and patient data model by GLEE has not yet solved the *curly braces problem*,[3] which refers to the hindrance of medical knowledge sharing caused by incompatible approaches to patient data representation [29]. However, the current approach can at least promote the standardization of patient data representation and thus move toward the long-term goal to share completely the medical knowledge that is embedded within guidelines.

Registration of clinical events and notification of clinical actions are implemented in GLEE using a similar approach, with the identification of a controlled terminology and the concept in that terminology corresponding to the event or the clinical action as the parameters in the communication between GLEE and the local environment.

9. Expression language and scheduling constraint specification language

The expression language is used in GLIF3 to encode decision criteria and patient states. Because an expression language is closely related to the data model that presupposes how the variables in an expression can be referenced, the standardization of an expression language partially relies on the standardization of patient data model. Due to the lack of a standard patient data model, standardization of the expression language is currently under development. Thus, GLIF3 does not assume a specific expression language. Instead, GLIF3 supports different expression languages, the appropriateness of which can be decided by guideline encoders. In the current implementation of GLEE, we use the Guideline Expression Language (GEL) [30,31], which is based on an extension of the logic expression used in the Arden Syntax, to encode decision criteria and patient state. The Backus Naur Form (BNF) notation for the syntax of the GEL language can be found at: http://guidelines.dbmi.columbia.edu/GLEE/GEL-BNF.html. It is important to note that the GEL parser is implemented as a separate package in GLEE, and thus it can be replaced by or complemented with parsers for other expression languages.[4] Again, this will facilitate the future maintenance of GLEE to support a standard expression language for clinical decision support.

The scheduling constraint specification language is used in GLIF3 to encode the continuation criterion of a synchronization step. In this language, the names of particular guideline steps are used as identifiers in the continuation criterion to represent the requirement on the completion of a synchronization step. The BNF notation for the syntax of the scheduling constraint specification language can be found at: http://guidelines.dbmi.columbia.edu/GLEE/SchedulingLanguageBNF.html.

10. Technical evaluation

We performed a study to evaluate the technical effectiveness of GLEE in terms of its functionality to execute guidelines encoded in the GLIF3 format. This study focused on the evaluation of GLEE's capability to

[3] Unless the controlled medical terminology and the patient data model used by GLEE are universally accepted, consistent interpretation of the encoded patient data still needs the mapping of the controlled medical terminologies and the patient data models at different local institutions. This becomes a local implementation task when integrating GLEE with a clinical information system.

[4] The GEL language is not object-oriented. As the virtual medical record will likely be an object-oriented data model, we are working on the development of an object-oriented expression language, GELLO [32].

correctly interpret the semantics of GLIF3 guidelines. Although real clinical data were used in the evaluation, this study did not aim to examine the use of GLEE to implement GLIF3 guidelines in clinical setting. The latter requires integration of GLEE with the clinical information system at a local institution to provide clinical decision support at the point of care, which constitutes one aspect of our future work.

We selected two guidelines as the subject guidelines in the evaluation: (1) the DTP series of the childhood immunization guidelines published by the CDC [33], which recommends the DTP vaccines due for an eligible child, and (2) the cough guideline published by the US Army [34], which recommends the possible diagnoses for patients with cough. These two guidelines were selected because their encoding require all the elements in the GLIF3 model. Thus, these two guidelines taken together can be used to examine whether GLEE can correctly handle each representation element in the GLIF3 model.

For the DTP immunization guideline, we reused 2007 patient cases that had been used previously in a clinical trial of the EzVac system [35], a computer-based immunization registry that implemented the same DTP immunization guideline but was not based on the GLIF3 model. For the cough guideline, domain experts manually created 20 typical patient cases, starting with a specific disease that may lead to cough and listing a set of typical symptoms, signs, and lab test results associated with that disease.

To evaluate its technical effectiveness, we used GLEE to execute the patient cases for each of the two subject guidelines. The appropriateness of the final recommendations generated by GLEE was used as the outcome variable. For the DTP immunization guideline, the recommendation outcome variable was the vaccines due for a child. For the cough guideline, the recommendation outcome variable was the possible diagnoses of cough. We used physicians' judgments as the gold standard in the evaluation. For the DTP immunization guideline, we compared the results generated by GLEE with those generated by the EzVac system, using the latter as an external reference to measure the performance of GLEE. For the cough guideline, we used the performance of GLEE on the first 10 cases as a reference to measure its performance on the last 10 cases. Since the first 10 cases were used to tune the encoding of the decision criteria of the cough guideline, GLEE's performance on these cases was used as the baseline data. To avoid possible bias due to the inconsistent understanding of the same DTP immunization guideline by different encoders, the author who was the primary developer of the DTP immunization guideline in the EzVac system (DW) encoded the same guideline in the GLIF3 format. To avoid possible bias due to the different styles of encoding, another author (MP) encoded the cough guideline in the GLIF3 format.

For the DTP immunization guideline, GLEE and EzVac generated consistent results for 1978 out of the 2007 cases (98.56%); while in the remaining 29 cases (1.44%), their execution results were inconsistent. To evaluate the appropriateness of the final recommendations, we gave all the 29 inconsistent cases and 20 cases that were randomly selected from the 1978 consistent cases to two attending physicians with sufficient familiarity with CDC's DTP childhood immunization guideline for the first round of the review. In this round, the two physician judges did not know the recommendations generated by GLEE and EzVac. Instead, they made their own judgments on the possible vaccines due for a specific patient case. These decisions were based solely on the case description of a patient's immunization history and other necessary information. The sensitivity and the specificity of GLEE in this round were 99.71% and 67.65%, respectively; and the sensitivity and the specificity of the EzVac system in this round were 99.43% and 67.48%, respectively.[5] Within the 49 cases that were reviewed by the two physicians in the first round, there were five cases on which the judgments by the physicians were different from the results generated by either GLEE or EzVac. As the EzVac system had already been validated in the previous clinical trial, the chance of an incorrect physician judgment is high when that judgment differs from either of the results generated by GLEE and EzVac. To improve the reliability of the physicians' judgments without excessive extra investment in time and human resources, only the above five cases were sent back to the physicians for a second review. This time the results generated by GLEE and EzVac were available to the physician judges. The sensitivity and the specificity of GLEE in the second round of the review were 99.80% and 80.74%, respectively; and the sensitivity and the specificity of the EzVac system in the second round of the review were 99.53% and 80.55%, respectively.

For the cough guideline, GLEE was used to execute 20 patient cases. For each case, it generated a set of possible diagnoses. To evaluate the appropriateness of these diagnoses, we gave all the 20 cases and their corresponding diagnosis sets to two attending physicians (different from the previous two physicians in the evaluation of the DTP immunization guideline) with sufficient familiarity with the cough guideline for a review. The percentage of the correct (the diagnosis is the original diagnosis from which the case was created), ac-

[5] Here, sensitivity = (total number of cases that were both detected by a computer-based guideline implementation system and judged by physicians as to have a vaccine due)/(total number of cases that were judged by physicians as to have a vaccine due), and specificity = (total number of cases that were both detected by a computer-based guideline implementation system and judged by physicians as to have no vaccine due)/(total number of cases that were judged by physicians as to have no vaccine due).

ceptable (the diagnosis is not the original diagnosis from which the case was created, but a reasonable diagnosis that may lead to the manifestation of the symptoms, signs, and lab test results shown in the case description), and wrong (the diagnosis is not the original diagnosis from which the case was created, and not a reasonable diagnosis that may lead to the manifestation of the symptoms, signs, and lab test results shown in the case description) diagnoses for case 1 to case 10 were 39%, 47%, and 14%, respectively; and the percentage of the correct, acceptable, and wrong diagnoses for case 11 to case 20 were 47%, 45%, and 8%, respectively. For all the 20 cases, the diagnosis set generated for a specific case contained the original diagnosis from which that case was created (for case 1 to case 10, this was realized through the tuning of the encoding; for case 11 to case 20, this was a natural result generated by GLEE).

The execution results of both guidelines have shown that the performance of GLEE in terms of the appropriateness of the final recommendations has reached the level of the reference systems against which it was being compared. Specifically, in the execution of the DTP immunization guideline, the sensitivity and the specificity of GLEE were at the same level of the EzVac system; in the execution of the cough guideline, the accuracy of GLEE when it was used to execute the last 10 cases was a little better than that when it was used to execute the first 10 cases. Analyses of the cases with clinically invalid results found that the problems were due to (1) errors in data preprocessing of the immunization guideline, (2) imperfect encoding of the cough guideline (due to the limitation of resources, when using the first 10 cases to tune the encoding of the cough guideline, we stopped the tuning for a case after verifying that the diagnosis from which the case was originally created had been included in the diagnosis set of that case generated by GLEE), (3) incorrect assumptions on the number of days in a month made by the GEL language (a problem that was inherited from the Arden Syntax), and (4) physicians' flexible interpretation of the decision criteria (e.g., while the DTP guideline recommended that the interval between dose 1 and dose 2 should be at least 6 weeks, one of the physician judge thought it was OK for 41 days). Only (3) was a problem of the GLEE system itself.

11. Discussion

GLEE is intended to be used as a modular interpreter of GLIF3 guidelines. Our results have shown that it can be used effectively for this purpose except on rare occasions due to a minor problem in the expression language used in the current implementation. However, even if an execution engine can correctly interpret guidelines, the recommendations it generates may still not be accepted by a clinician. The results of our evaluation have shown that poor data quality, imperfect encoding of a guideline, and clinicians' flexible interpretation of guideline may all lead to the unacceptability of a guideline's advice, even when otherwise the guideline would have been correctly executed. Thus, we believe the execution flexibility provided by GLEE is especially important to address this issue when implementing guidelines in clinical settings. With such flexibility, even if clinicians reject an irrelevant or inappropriate suggestion when applying a guideline to a patient, they do not have to abandon the application of the entire guideline. Instead, they can always come back to the guideline and start a relevant step at a later time when they think it is appropriate. In this way, GLEE may overcome the limitations of previous approaches that depend on (1) a guideline encoder's enumeration of all possible clinical states of a patient during the application of a specific guideline, (2) high quality of clinical data, and (3) clinicians' stringent interpretation of the guideline. Comparing to the approach used by GUIDE that can only handle a limited set of exceptions [36], the execution flow of GLEE can be arbitrarily changed by users as they judge as appropriate. Further evaluation of GLEE on its use for this purpose will be necessary after it has been integrated with the clinical information system at one or more local institutions to provide clinical decision support in practice.

Integration of decision support within clinicians' workflow is a critical factor for its success [18]. Thus, we believe a standard interface between a guideline execution system and associated clinical applications is important for guideline implementation in clinical settings. The interface to the clinical applications provided by the GLEE system is intended to facilitate this integration and to promote the use of guidelines in clinical practice. Nevertheless, we do not exclude the possibility that a local system can provide its own methods of communication and presentation for alerts and reminders. GLEE thus provides the messaging function at the backend so that it can be used in this case as an alternative to facilitate the communication and integration. Further evaluation on this capability of GLEE needs to be performed in clinical settings.

Finally, the interface between the guideline representation model and other parts of the execution engine facilitates the maintenance of the execution system itself, including the evolution of the representation model and the generalization of the execution model. Further discussion on this topic is beyond the scope of this paper but can be found elsewhere [37].

Due to the restriction of available resources, the evaluation of GLEE was performed in a lab setting on a limited scale. In a production environment, additional efforts should be placed into the guideline encoding process to maximize its correctness. In addition, when future evaluations on GLEE's capabilities are

performed, more clinician judges should be included and trained to enhance the reliability of their judgments.

We are now working on the integration of GLEE with the clinical information system at New York-Presbyterian Hospital. Once this integration is completed, we will perform further studies on GLEE's use in clinical settings, including the impact of its execution flexibility on clinicians' use of guidelines.

12. Summary

We have developed the GLEE system to balance the requirements of guideline sharing, execution flexibility, and system maintenance. The technical evaluation of GLEE has shown that it can be used effectively for execution of guidelines encoded in the GLIF3 format. GLEE's use in clinical settings needs to be evaluated in the future. We understand, however, that wide acceptance of a guideline system in clinical practice depends on many other factors, such as the development of a widely-accepted standard patient-data model and the in-depth understanding of local adaptation of guidelines. These issues are not addressed in our current work but help to define the challenges for the future. On the other hand, we believe that the modular design of GLEE will facilitate adoption and use of such standards for guideline-based decision support when a consensus develops.

Acknowledgments

This work has been supported by the Grant LM06594 to the InterMed Collaboratory from the National Library of Medicine, the US Army, and the Agency for Healthcare Research and Quality. We thank Drs. Davis Bu, Michael Cantor, Giora Landesberg, and Eitan Lunenfeld for their helps on the evaluation of GLEE.

References

[1] Shortliffe EH, Patel VL, Cimino JJ, Barnett GO, Greenes RA. A study of collaboration among medical informatics research laboratories. Artif Intell Med 1998;12(2):97–123.
[2] Greenes RA, Peleg M, Boxwala A, Tu S, Patel V, Shortliffe EH. Sharable computer-based clinical practice guidelines: rationale, obstacles, approaches, and prospects. Medinfo 2001;10(Pt 1):201–5.
[3] Patel VL, Branch T, Wang D, Peleg M, Boxwala A. Analysis of the process of encoding guidelines: a comparison of GLIF2 and GLIF3. Methods Inf Med 2002;41(2):105–13.
[4] Boxwala AA, Tu S, Peleg M, Zeng Q, Ogunyemi O, Greenes RA, et al. Toward a representation format for sharable clinical guidelines. J Biomed Inform 2001;34(3):157–69.
[5] Maviglia SM, Zielstorff RD, Paterno M, Teich JM, Bates DW, Kuperman GJ. Automating complex guidelines for chronic disease: lessons learned. J Am Med Inform Assoc 2003;10(2):154–65.
[6] Tu SW, Musen MA. From guideline modeling to guideline executions: defining guideline-based decision-support services. Proc AMIA Symp 2000:863–7.
[7] Fox J, Das S. Safe and sound: artificial intelligence in hazardous applications. Menlo Park, CA: AAAI Press/The MIT Press; 2000.
[8] De Clercq PA, Blom JA, Hasman A, Korsten HH. GASTON: an architecture for the acquisition and execution of clinical guideline-application tasks. Med Inform Internet Med 2000;25(4):247–63.
[9] Dubey AK, Chueh HC. An XML-based format for guideline interchange and execution. Proc AMIA Symp 2000:205–9.
[10] Boxwala AA, Greenes RA, Deibel SR. Architecture for a multipurpose guideline execution engine. Proc AMIA Symp 1999:701–5.
[11] Cabana MD, Rand CS, Powe NR, Wu AW, Wilson MH, Abboud PA, et al. Why don't physicians follow clinical practice guidelines? A framework for improvement. JAMA 1999;282(15):1458–65.
[12] Ohno-Machado L, Gennari JH, Murphy SN, Jain NL, Tu SW, Oliver DE, et al. The guideline interchange format: a model for representing guidelines. J Am Med Inform Assoc 1998;5(4):357–72.
[13] Peleg M, Boxwala AA, Tu S, Zeng Q, Ogunyemi O, Wang D, et al. The intermed approach to sharable computer-interpretable guidelines: a review. J Am Med Inform Assoc 2004;11(1):1–10.
[14] InterMed Collaboratory. Guideline Interchange Format 3.4 Technical Specification. 2000. Available at: http://www.smi.stanford.edu/projects/intermed-web/guidelines/GLIF_TECH_SPEC.pdf. Accessed March 18, 2003.
[15] Resource Description Framework. Available at: http://www.w3.org/RDF/. Accessed March 18, 2003.
[16] Wang D, Shortliffe EH. GLEE—a model-driven execution system for computer-based implementation of clinical practice guidelines. Proc AMIA Symp 2002;1:855–9.
[17] Zielstorff RD. Online practice guidelines: issues, obstacles, and future prospects. J Am Med Inform Assoc 1998;5(3):227–36.
[18] Tierney WM, Overhage JM, McDonald CJ. Computerizing guidelines: factors for success. Proc AMIA Annu Fall Symp 1996;1:459–62.
[19] Musen MA, Fergerson RW, Grosso WE, Noy NF, Crubezy M, Gennari JH. Component-based support for building knowledge-acquisition systems. Conference on Intelligent Information Processing (IIP 2000) of the International Federation for Information Processing World Computer Congress (WCC 2000). 2000.
[20] Tu SW, Musen MA. A flexible approach to guideline modeling. Proc AMIA Symp 1999:420–4.
[21] Johnson PD, Tu S, Booth N, Sugden B, Purves IN. Using scenarios in chronic disease management guidelines for primary care. Proc AMIA Symp 2000:389–93.
[22] Extensible Markup Language. Available at: http://www.w3.org/XML/. Accessed March 18, 2003.
[23] Hripcsak G, Clayton PD, Jenders RA, Cimino JJ, Johnson SB. Design of a clinical event monitor. Comput Biomed Res 1996;29(3):194–221.
[24] Schadow G, Russler DC, McDonald CJ. Conceptual alignment of electronic health record data with guideline and workflow knowledge. Int J Med Inf 2001;64(2–3):259–74.
[25] Stearns MQ, Price C, Spackman KA, Wang AY. SNOMED clinical terms: overview of the development process and project status. Proc AMIA Symp 2001;1:662–6.
[26] Huff SM, Rocha RA, McDonald CJ, De Moor GJ, Fierr T, Bidgood Jr WD, et al. Development of the logical observation

identifier names and codes (LOINC) vocabulary. J Am Med Inform Assoc 1998;5(3):276–92.
[27] Zeng Q, Tu S, Boxwala AA, Peleg M, Greenes RA, Shortliffe EH. A three-layer domain ontology for guideline representation and sharing. Proc AMIA Symp 2000;1:1164.
[28] Johnson PD, Tu SW, Musen MA, Purves I. A virtual medical record for guideline-based decision support. Proc AMIA Symp 2001;1:294–8.
[29] Pryor TA, Hripcsak G. Sharing MLM's: an experiment between Columbia-Presbyterian and LDS hospital. Proc Annu Symp Comput Appl Med Care 1993:399–403.
[30] Peleg M, Ogunyemi O, Tu S, Boxwala AA, Zeng Q, Greenes RA, et al. Using features of arden syntax with object-oriented medical data models for guideline modeling. Proc AMIA Symp 2001:523–7.
[31] Ogunyemi O. The guideline expression language (GEL) user's guide. Technical Report, DSG-TR-2000-001, 2000, Decision Systems Group, Boston, MA.
[32] Sordo M, Ogunyemi O, Boxwala AA, Greenes RA. Software specification for GELLO: an object-oriented query and expression language for clinical decision support. Technical Report, DSG-TR-2003-02, 2003, Decision Systems Group, Boston, MA.
[33] Centers for Disease Control and Prevention. General recommendations on immunization recommendations of the Advisory Committee on Immunization Practices (ACIP). MMWR Morb Mortal Wkly Rep 1994;43(RR01):1–38.
[34] Whitlock W. "Cough", Army Cough Guideline: Special Operations Forces Medical Handbook. Teton NewMedia, 2001.
[35] Wang D, Jenders RA, Dasgupta B. Translating national childhood immunization guidelines to a computer-based reminder recall system within an immunization registry. Proc AMIA Annu Symp 1999:994–8.
[36] Quaglini S, Stefanelli M, Lanzola G, Caporusso V, Panzarasa S. Flexible guideline-based patient careflow systems. Artif Intell Med 2001;22(1):65–80.
[37] Wang D, Peleg M, Bu D, Cantor M, Landesberg G, Lunenfeld E, et al. GESDOR—a generic execution model for sharing of computer-interpretable clinical practice guidelines. Proc AMIA Symp 2003:694–8.